Technical Cybersecurity

Unix Passwords

Let’s look at UNIX passwords.

CRYPT(3) FUNCTION

- Most password implementations use this
- Implementation varies from system to system

HOW CAN YOU TELL?

- MD5: Password starts with ‘1’
- BSDi DES: … starts with ‘_’
- Blowfish: … starts with ‘2’ or ‘$2a$’
- SHA-256: …starts with ‘5’
- SHA-512: …starts with ‘6’
- DES: no decorations!

DES

RARE TODAY

- DES is not a strong encryption algorithm, broken in ’97

HOW DID IT WORK?

1. Truncate or pad password to 8 characters
2. Compress to 7-bit chars; this yields a 56-bit bitstring
3. Using the bitstring as a key, encrypt a (usually zero) block N (usually 25) times, using a 12-bit salt
4. Base64 encode the result

MD5

Still common, especially in IoT devices

- But MD5 (as a hashing algorithm) is considered broken as well

Process

1. Prepend salt to password and hash
2. Prepend original password and salt to hash from step (1), and hash
3. Repeat (1) and (2) for multiple rounds, changing order of elements for up to 1,000 rounds

SHA is similar

- But different algorithm and 5,000 rounds by default

Final Representation

Stored in password files

- <prepend token><salt>$<hash>

Examples

- **$1$salty$hashy**: MD5, *salt*: salty, *hash*: hashy
- **$5$no_sugar$this_is_the_hash**: SHA-256, *salt*: no_sugar, *hash*: this_is_the_hash

How are these attacked?